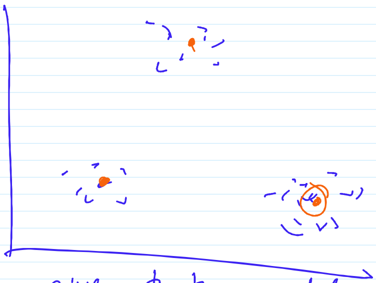
K Means Clustering → whatsapp group
(No of clusters)

Clusters → high Intercluster
→ low Intracluster distance } → $DI = \frac{\min(Int)}{\max(Intra)}$.

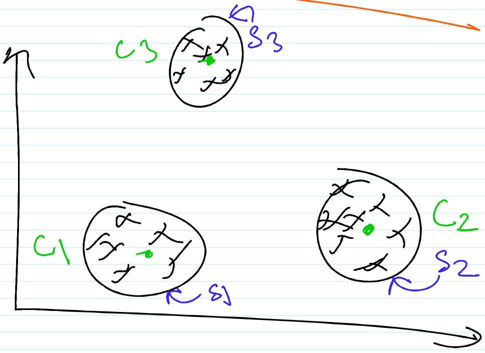
KMeans Clustering
(Algo)
→ simple
→ K = no of clusters



#

Hyper parameter ⇒ which you give to the model

→



$S_1 \cup S_2 \cup S_3$ = Data (All)

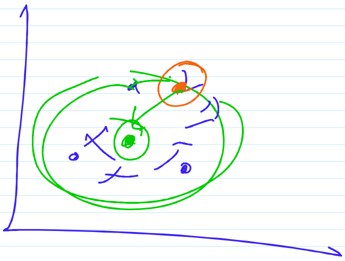
$S_1 \cap S_2 \cap S_3 = \phi$

Centroid

	x_1	x_2	x_3
1 →	2	3	5
2 →	3	8	9
3 →	4	9	10

$\frac{2+3+4}{3}$ $\frac{3+8+9}{3}$ $\frac{5+9+10}{3}$

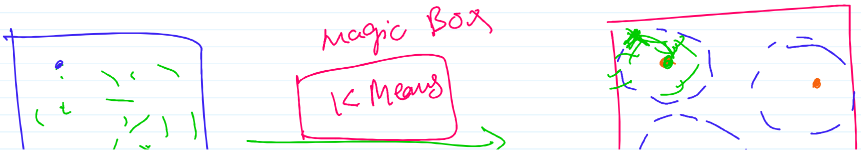
plot →



[One point in a cluster can only belong to one cluster]

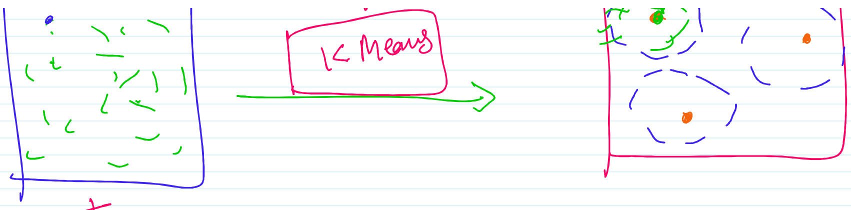
Goal → We have to find $\underbrace{C_1 \; C_2 \; --- \; C_K}$
(K-clusters)

Algo

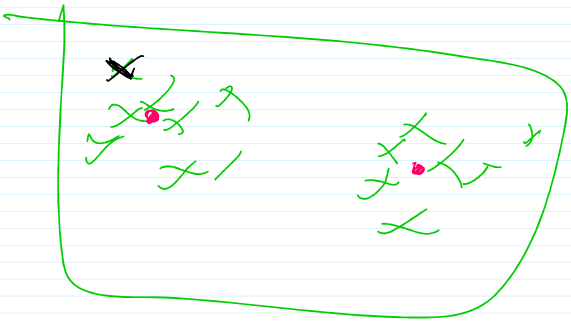


(Assign each

Algo



+ (K=3)



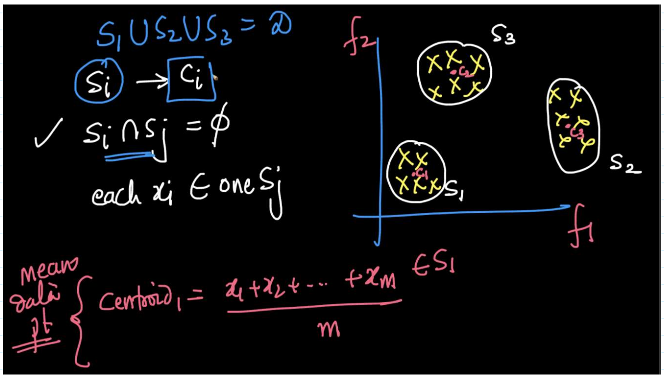
Mathematical Formulations

(k → we will give)

Goal : Find C_1 C_2 C_3 - - - - C_K

s.t ① Intercluster Distance is maximised

② minimise Intercluster distance



$S_1 \cup S_2 \cup S_3 = \mathcal{D}$

$\boxed{S_i} \rightarrow \boxed{C_i}$

✓ $S_i \cap S_j = \phi$

each $x_i \in$ one S_j

mean data pt $\left\{ \text{Centroid}_1 = \frac{x_1 + x_2 + \cdots + x_m}{m} \in S_1 \right.$

Optimisation → (NP hard) → exponential time complexity

(#) Each data pt. is assigned only to one cluster

Lloyd's Algo

(Approximate)

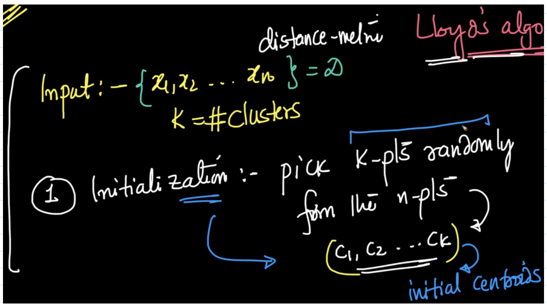
(*) Lloyds Algoritm → only 3 step Process

① Random initialise points (k points)

② Assign each point to nearest cluster

③ Re compute / Update the centroid

① Randomly initialise points (K) points →



distance-metric

Lloyd's algo

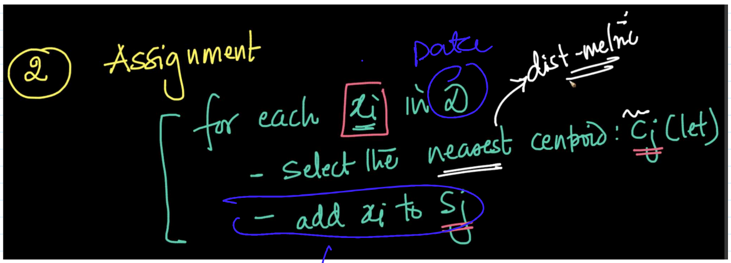
Input :- $\{x_1, x_2 \ldots x_n\} = \mathcal{D}$

$K = \#$clusters

① Initialization :- pick K-pts randomly from the n-pts

$(c_1, c_2 \ldots c_K)$ initial centroids

② Calculate distance of [x_i $i=1$ to N] from each centroid & then assign the label

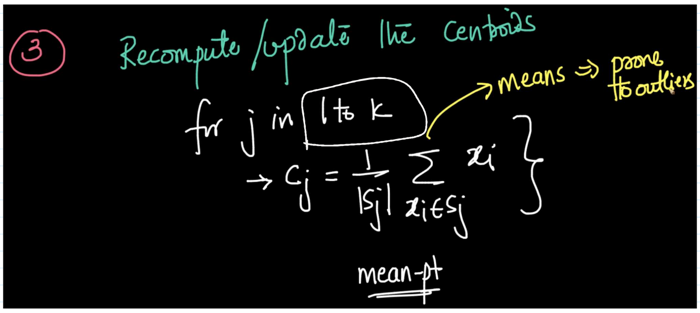


② Assignment

```
Data                                   dist-metric
for each [x_i] in D
    - select the nearest centroid : c̃_j (let)
    - add x_i to S_j
```

↳ assign that x_i point to S_j cluster

Update each of the (K) clusters ⇒



③ Recompute/update the centroids

```
for j in [1 to k]        → means ⇒ prone to outliers
    → $c_j = \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i$ }
```

mean-pt

K Means (Lloyds Algo) Scratch Implementation (k=3)

① Random initals^n → np.random.choice(   )

② Dist (x_i, with each centroid)

(Euclidian)

$$\begin{tikzcd} & k_1 \\ x_i \arrow[ur] \arrow[r] \arrow[dr] & k_2 \text{ (min)} \\ & k_3 \end{tikzcd}$$

⇒ $x_i \rightarrow k_2$

③ Recompute centroids $\left(\frac{\Sigma x}{n}\right) \Big/ \left(\frac{\Sigma y}{n}\right)$

k → hyperparameters (you have to give k)

How to find the right k ?

* (Business Sense)
  - Fashion tag its products
  - Car Rental Company
    - ↳ Car A (Hatchback)
    - (SUV)

* (Elbow Method)
  - ↳ General concept in clustering : WCSS (within clusters sums of square)

products

Shirt Jeans Night Gown

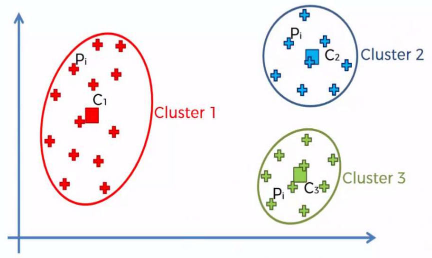
⇒ (K=3)

Car A (Hatchback)

Car B (SUV)

Car C (Luxury)

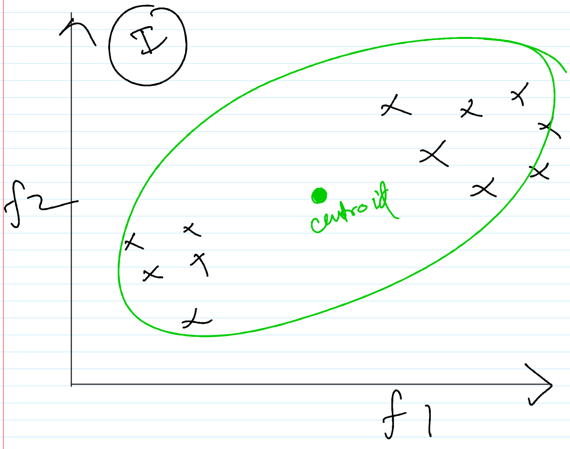
K=3

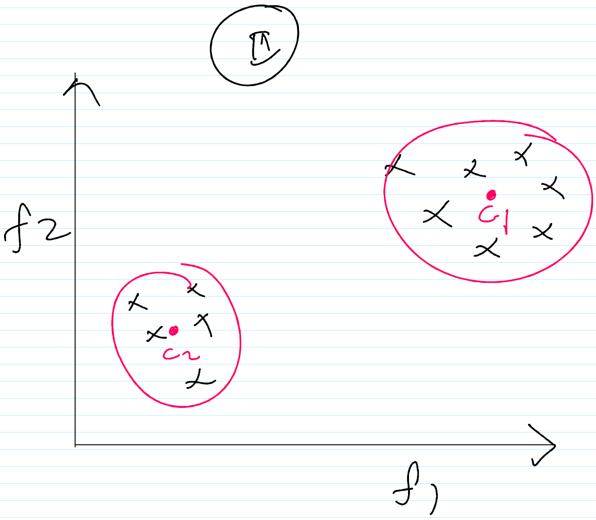


$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

(Effect of K on $\underbrace{\text{WCSS}}$)

no of clusters

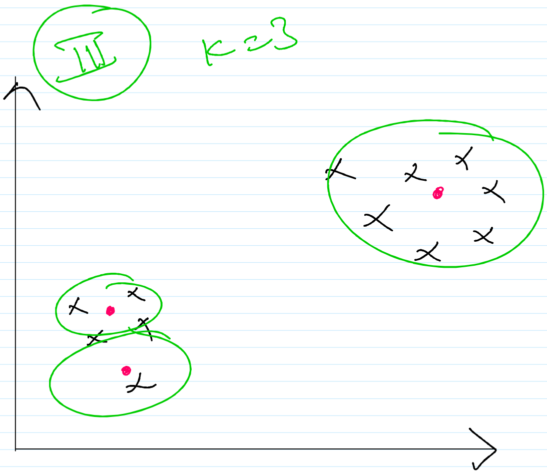




(a) WCSS ① > 2

(b) ① < ②

(c) ① = ②



K=3

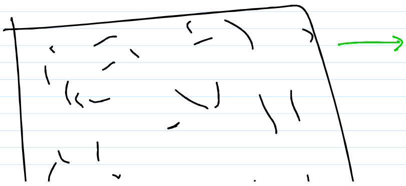
[As K ↑ my WCSS decreases]

Quiz when WCSS = 0 ?

① K = ?

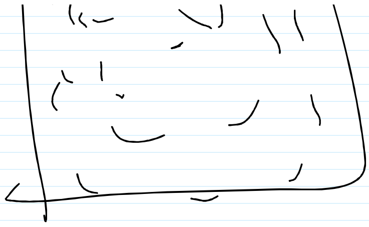
n = (100 data points)

Elbow Method

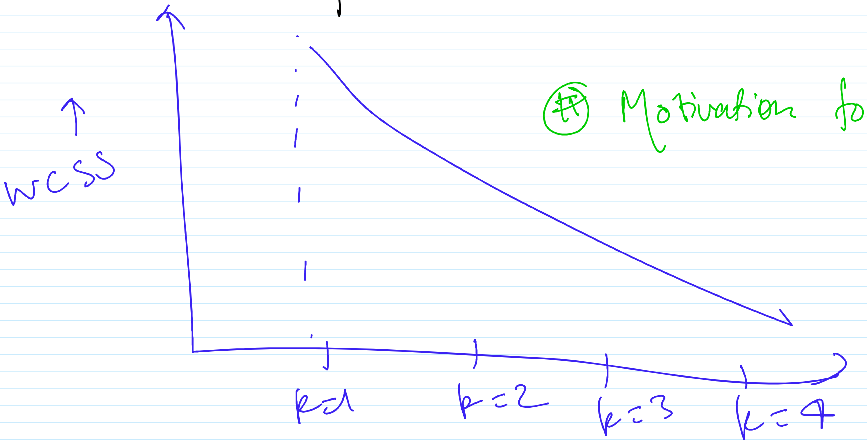


→ WCSS for K=1 to say (5)

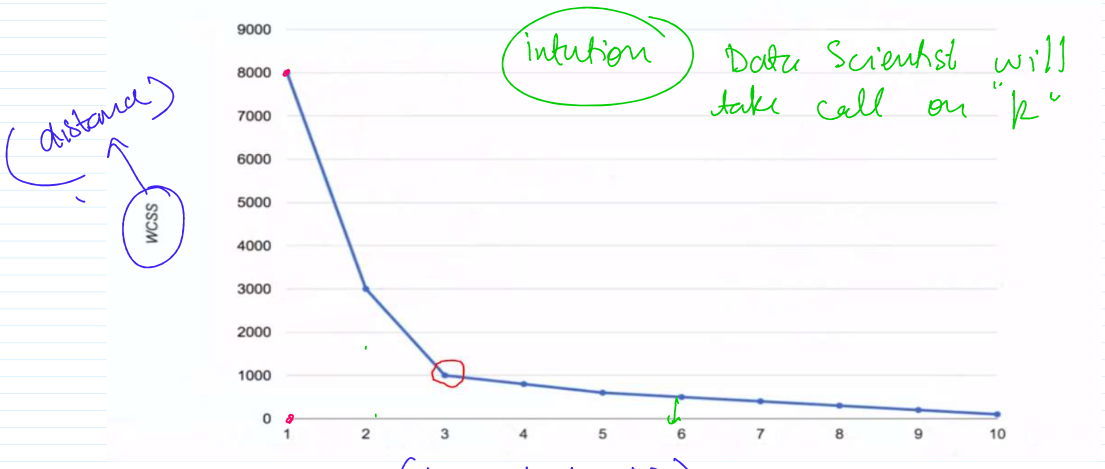
(3—4)



(3 — 4)



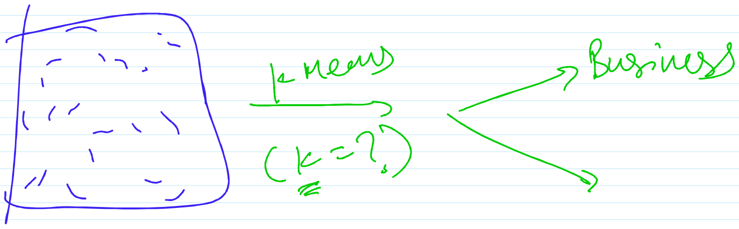
⊕ Motivation for finding good k ⇒ Slope at each point



intution

Data Scientist will take call on "k"

*(k = 1 to 10)



⊕ Trap Points → {Edge Cases of Algorithm}

Low Dimension → Use Eucledians

Low – Med → Manhattan

– high → cosine